The Tradeoffs Between Open and Traditional Relation Extraction

Michele Banko and Oren Etzioni
Turing Center
University of Washington
Computer Science and Engineering
Box 352350
Seattle, WA 98195, USA
banko,etzioni@cs.washington.edu

Abstract

Traditional Information Extraction (IE) takes a relation name and hand-tagged examples of that relation as input. Open IE is a *relation-independent* extraction paradigm that is tailored to massive and heterogeneous corpora such as the Web. An Open IE system extracts a diverse set of relational tuples from text without *any* relation-specific input. How is Open IE possible? We analyze a sample of English sentences to demonstrate that numerous relationships are expressed using a compact set of relation-independent lexico-syntactic patterns, which can be learned by an Open IE system.

What are the tradeoffs between Open IE and traditional IE? We consider this question in the context of two tasks. First, when the number of relations is massive, and the relations themselves are not pre-specified, we argue that Open IE is necessary. We then present a new model for Open IE called O-CRF and show that it achieves increased precision and nearly double the recall than the model employed by TEXTRUNNER, the previous state-of-the-art Open IE system. Second, when the number of target relations is small, and their names are known in advance, we show that O-CRF is able to match the precision of a traditional extraction system, though at substantially lower recall. Finally, we show how to combine the two types of systems into a hybrid that achieves higher precision than a traditional extractor, with comparable recall.

1 Introduction

Relation Extraction (RE) is the task of recognizing the assertion of a particular relationship between two or more entities in text. Typically, the target relation (*e.g.*, seminar location) is given to the RE system as input along with hand-crafted extraction patterns or patterns learned from hand-labeled training examples (Brin, 1998; Riloff and Jones, 1999; Agichtein and Gravano, 2000). Such inputs are *specific* to the target relation. Shifting to a new relation requires a person to manually create new extraction patterns or to specify new training examples by hand. This manual labor scales linearly with the number of target relations.

In 2007, we introduced a new approach to the RE task, called Open Information Extraction (Open IE), which scales RE to the Web. An Open IE system extracts a diverse set of relational tuples without requiring *any* relation-specific human input. Open IE's extraction process is linear in the number of documents in the corpus, and constant in the number of relations. Thus, Open IE is ideally suited to corpora such as the Web, where the target relations are not known in advance, and their number is massive.

The relationship between standard RE systems and the new Open IE paradigm is analogous to the relationship between lexicalized and unlexicalized parsers. Statistical parsers are usually *lexicalized* (*i.e.* they make parsing decisions based on n-gram statistics computed for specific lexemes). However, Klein and Manning (2003) showed that *unlexicalized* parsers are more accurate than previously believed, and can be learned in an unsupervised manner. Klein and Manning analyze the tradeoffs between the two approaches to parsing and argue that state-of-the-art parsing will benefit from employing both approaches in concert. In this paper, we examine the tradeoffs between relation-specific ("lexicalized") extraction and relation-independent ("unlexicalized") extraction and reach an analogous conclusion.

Is it, in fact, possible to learn relation-independent extraction patterns? What do they look like? We first consider the task of open extraction, in which the goal is to extract relationships from text when their number is large and identity unknown. We then consider the targeted extraction task, in which the goal is to locate instances of a known relation. How does the precision and recall of Open IE compare with that of relation-specific extraction? Is it possible to combine Open IE with a "lexicalized" RE system to improve performance? This paper addresses the questions raised above and makes the following contributions:

- We present O-CRF, a new Open IE system that uses Conditional Random Fields, and demonstrate its ability to extract a variety of relations with a precision of 88.3% and recall of 45.2%. We compare O-CRF to O-NB, the extraction model previously used by TEXTRUNNER (Banko et al., 2007), a state-of-the-art Open IE system. We show that O-CRF achieves a relative gain in F-measure of 63% over O-NB.

- We provide a corpus-based characterization of how binary relationships are expressed in English to demonstrate that learning a relation-independent extractor is feasible, at least for the English language.

- In the targeted extraction case, we compare the performance of O-CRF to a traditional RE system and find that without any relation-specific input, O-CRF obtains the same precision with lower recall compared to a lexicalized extractor trained using hundreds, and sometimes thousands, of labeled examples per relation.

- We present H-CRF, an ensemble-based extractor that learns to combine the output of the lexicalized and unlexicalized RE systems and achieves a 10% relative increase in precision with comparable recall over traditional RE.

The remainder of this paper is organized as follows. Section 2 assesses the promise of relation-independent extraction for the English language by characterizing how a sample of relations is expressed in text. Section 3 describes O-CRF, a new Open IE system, as well as R1-CRF, a standard RE system; a hybrid RE system is then presented in Section 4. Section 5 reports on our experimental results. Section 6 considers related work, which is then followed by a discussion of future work.

2 The Nature of Relations in English

How are relationships expressed in English sentences? In this section, we show that many relationships are consistently expressed using a compact set of relation-independent lexico-syntactic patterns, and quantify their frequency based on a sample of 500 sentences selected at random from an IE training corpus developed by (Bunescu and Mooney, 2007).[1] This observation helps to explain the success of open relation extraction, which learns a relation-independent extraction model as described in Section 3.1.

Previous work has noted that distinguished relations, such as hypernymy (is-a) and meronymy (part-whole), are often expressed using a small number of lexico-syntactic patterns (Hearst, 1992). The manual identification of these patterns inspired a body of work in which this initial set of extraction patterns is used to seed a bootstrapping process that automatically acquires additional patterns for is-a or part-whole relations (Etzioni et al., 2005; Snow et al., 2005; Girju et al., 2006), It is quite natural then to consider whether the same can be done for *all* binary relationships.

To characterize how binary relationships are expressed, one of the authors of this paper carefully studied the labeled relation instances and produced a lexico-syntactic pattern that captured the relation for each instance. Interestingly, we found that 95% of the patterns could be grouped into the categories listed in Table 1. Note, however, that the patterns

[1] For simplicity, we restrict our study to binary relationships.

Relative Frequency	Category	Simplified Lexico-Syntactic Pattern
37.8	Verb	E_1 Verb E_2 *X established Y*
22.8	Noun+Prep	E_1 NP Prep E_2 *X settlement with Y*
16.0	Verb+Prep	E_1 Verb Prep E_2 *X moved to Y*
9.4	Infinitive	E_1 to Verb E_2 *X plans to acquire Y*
5.2	Modifier	E_1 Verb E_2 Noun *X is Y winner*
1.8	Coordinate$_n$	E_1 (and|,|-|:) E_2 NP *X-Y deal*
1.0	Coordinate$_v$	E_1 (and|,) E_2 Verb *X , Y merge*
0.8	Appositive	E_1 NP (:|,)? E_2 *X hometown : Y*

Table 1: Taxonomy of Binary Relationships: Nearly 95% of 500 randomly selected sentences belongs to one of the eight categories above.

shown in Table 1 are greatly simplified by omitting the exact conditions under which they will reliably produce a correct extraction. For instance, while many relationships are indicated strictly by a verb, detailed contextual cues are required to determine, exactly which, if any, verb observed in the context of two entities is indicative of a relationship between them. In the next section, we show how we can use a Conditional Random Field, a model that can be described as a finite state machine with weighted transitions, to learn a model of how binary relationships are expressed in English.

3 Relation Extraction

Given a relation name, labeled examples of the relation, and a corpus, traditional Relation Extraction (RE) systems output instances of the given relation found in the corpus. In the open extraction task, relation names are not known in advance. The sole input to an Open IE system is a corpus, along with a small set of relation-independent heuristics, which are used to learn a general model of extraction for all relations at once.

The task of open extraction is notably more difficult than the traditional formulation of RE for several reasons. First, traditional RE systems do not attempt to extract the text that signifies a relation in a sentence, since the relation name is given. In contrast, an Open IE system has to locate both the set of entities believed to participate in a relation, and the salient textual cues that indicate the relation among them. Knowledge extracted by an open system takes the form of relational tuples $(r, e_1, \ldots, e_n)$ that contain two or more entities $e_1, \ldots, e_n$, and r, the name of the relationship among them. For example, from the sentence, "Microsoft is headquartered in beautiful Redmond", we expect to extract (*is headquartered in*, *Microsoft*, *Redmond*). Moreover, following extraction, the system must identify exactly which relation strings r correspond to a general relation of interest. To ensure high-levels of coverage on a *per-relation* basis, we need, for example to deduce that " 's headquarters in", "is headquartered in" and "is based in" are different ways of expressing HEADQUARTERS(X,Y).

Second, a relation-independent extraction process makes it difficult to leverage the full set of features typically used when performing extraction one relation at a time. For instance, the presence of the words *company* and *headquarters* will be useful in detecting instances of the HEADQUARTERS(X,Y) relation, but are not useful features for identifying relations in general. Finally, RE systems typically use named-entity types as a guide (*e.g.*, the second argument to HEADQUARTERS should be a LOCATION). In Open IE, the relations are not known in advance, and neither are their argument types.

The unique nature of the open extraction task has led us to develop O-CRF, an open extraction system that uses the power of graphical models to identify relations in text. The remainder of this section describes O-CRF, and compares it to the extraction model employed by TEXTRUNNER, the first Open IE system (Banko et al., 2007). We then describe R1-CRF, a RE system that can be applied in a typical one-relation-at-a-time setting.

3.1 Open Extraction with Conditional Random Fields

TEXTRUNNER initially treated Open IE as a classification problem, using a Naive Bayes classifier to

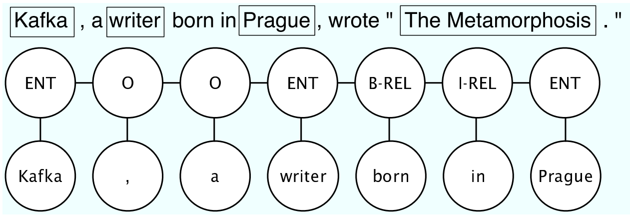

Figure 1: Relation Extraction as Sequence Labeling: A CRF is used to identify the relationship, *born in*, between *Kafka* and *Prague*

predict whether heuristically-chosen tokens between two entities indicated a relationship or not. For the remainder of this paper, we refer to this model as O-NB. Whereas classifiers predict the label of a single variable, graphical models model multiple, interdependent variables. Conditional Random Fields (CRFs) (Lafferty et al., 2001), are undirected graphical models trained to maximize the conditional probability of a finite set of labels Y given a set of input observations X. By making a first-order Markov assumption about the dependencies among the output variables Y, and arranging variables sequentially in a linear chain, RE can be treated as a sequence labeling problem. Linear-chain CRFs have been applied to a variety of sequential text processing tasks including named-entity recognition, part-of-speech tagging, word segmentation, semantic role identification, and recently relation extraction (Culotta et al., 2006).

3.1.1 Training

As with O-NB, O-CRF's training process is self-supervised. O-CRF applies a handful of relation-independent heuristics to the PennTreebank and obtains a set of labeled examples in the form of relational tuples. The heuristics were designed to capture dependencies typically obtained via syntactic parsing and semantic role labelling. For example, a heuristic used to identify positive examples is the extraction of noun phrases participating in a subject-verb-object relationship, *e.g.*, "<Einstein> received <the Nobel Prize> in 1921." An example of a heuristic that locates negative examples is the extraction of objects that cross the boundary of an adverbial clause, *e.g.* "He studied <Einstein's work> when visiting <Germany>."

The resulting set of labeled examples are described using features that can be extracted without syntactic or semantic analysis and used to train a CRF, a sequence model that learns to identify spans of tokens believed to indicate explicit mentions of relationships between entities.

O-CRF first applies a phrase chunker to each document, and treats the identified noun phrases as candidate entities for extraction. Each pair of entities appearing no more than a maximum number of words apart and their surrounding context are considered as possible evidence for RE. The entity pair serves to anchor each end of a linear-chain CRF, and both entities in the pair are assigned a fixed label of ENT. Tokens in the surrounding context are treated as possible textual cues that indicate a relation, and can be assigned one of the following labels: B-REL, indicating the start of a relation, I-REL, indicating the continuation of a predicted relation, or O, indicating the token is not believed to be part of an explicit relationship. An illustration is given in Figure 1.

The set of features used by O-CRF is largely similar to those used by O-NB and other state-of-the-art relation extraction systems, They include part-of-speech tags (predicted using a separately trained maximum-entropy model), regular expressions (*e.g.*detecting capitalization, punctuation, *etc.*), context words, and conjunctions of features occurring in adjacent positions within six words to the left and six words to the right of the current word. A unique aspect of O-CRF is that O-CRF uses context words belonging only to *closed classes* (*e.g.* prepositions and determiners) but not function words such as verbs or nouns. Thus, unlike most RE systems, O-CRF does not try to recognize semantic classes of entities.

O-CRF has a number of limitations, most of which are shared with other systems that perform extraction from natural language text. First, O-CRF only extracts relations that are explicitly mentioned in the text; implicit relationships that could inferred from the text would need to be inferred from O-CRF extractions. Second, O-CRF focuses on relationships that are primarily word-based, and not indicated solely from punctuation or document-level features. Finally, relations must occur between entity names within the same sentence.

O-CRF was built using the CRF implementation provided by MALLET (McCallum, 2002), as well as part-of-speech tagging and phrase-chunking tools available from OPENNLP.[2]

3.1.2 Extraction

Given an input corpus, O-CRF makes a single pass over the data, and performs entity identification using a phrase chunker. The CRF is then used to label instances relations for each possible entity pair, subject to the constraints mentioned previously.

Following extraction, O-CRF applies the RESOLVER algorithm (Yates and Etzioni, 2007) to find relation synonyms, the various ways in which a relation is expressed in text. RESOLVER uses a probabilistic model to predict if two strings refer to the same item, based on relational features, in an unsupervised manner. In Section 5.2 we report that RESOLVER boosts the recall of O-CRF by 50%.

3.2 Relation-Specific Extraction

To compare the behavior of open, or "unlexicalized," extraction to relation-specific, or "lexicalized" extraction, we developed a CRF-based extractor under the traditional RE paradigm. We refer to this system as R1-CRF.

Although the graphical structure of R1-CRF is the same as O-CRF R1-CRF differs in a few ways. A given relation R is specified *a priori*, and R1-CRF is trained from hand-labeled positive and negative instances of R. The extractor is also permitted to use all lexical features, and is not restricted to closed-class words as is O-CRF. Since R is known in advance, if R1-CRF outputs a tuple at extraction time, the tuple is believed to be an instance of R.

4 Hybrid Relation Extraction

Since O-CRF and R1-CRF have complementary views of the extraction process, it is natural to wonder whether they can be combined to produce a more powerful extractor. In a variety of machine learning settings, the use of an ensemble of diverse classifiers during prediction has been observed to yield higher levels of performance compared to individual algorithms. We now describe an ensemble-based or *hybrid* approach to RE that leverages the different views offered by open, self-supervised extraction in O-CRF, and lexicalized, supervised extraction in R1-CRF.

4.1 Stacking

Stacked generalization, or *stacking*, (Wolpert, 1992), is an ensemble-based framework in which the goal is learn a meta-classifier from the output of several base-level classifiers. The training set used to train the meta-classifier is generated using a leave-one-out procedure: for each base-level algorithm, a classifier is trained from all but one training example and then used to generate a prediction for the left-out example. The meta-classifier is trained using the predictions of the base-level classifiers as features, and the true label as given by the training data.

Previous studies (Ting and Witten, 1999; Zenko and Dzeroski, 2002; Sigletos et al., 2005) have shown that the probabilities of each class value as estimated by each base-level algorithm are effective features when training meta-learners. Stacking was shown to be consistently more effective than voting, another popular ensemble-based method in which the outputs of the base-classifiers are combined either through majority vote or by taking the class value with the highest average probability.

4.2 Stacked Relation Extraction

We used the stacking methodology to build an ensemble-based extractor, referred to as H-CRF. Treating the output of an O-CRF and R1-CRF as black boxes, H-CRF learns to predict which, if any, tokens found between a pair of entities (e_1, e_2), indicates a relationship. Due to the sequential nature of our RE task, H-CRF employs a CRF as the meta-learner, as opposed to a decision tree or regression-based classifier.

H-CRF uses the probability distribution over the set of possible labels according to each O-CRF and R1-CRF as features. To obtain the probability at each position of a linear-chain CRF, the constrained forward-backward technique described in (Culotta and McCallum, 2004) is used. H-CRF also computes the Monge Elkan distance (Monge and Elkan, 1996) between the relations predicted by O-CRF and R1-CRF and includes the result in the feature set. An additional meta-feature utilized by H-CRF indicates whether either or both base extractors return "no relation" for a given pair of entities. In addition to

[2] http://opennlp.sourceforge.net

Category	O-CRF			O-NB		
	P	**R**	**F1**	**P**	**R**	**F1**
Verb	93.9	65.1	76.9	100	38.6	55.7
Noun+Prep	89.1	36.0	51.3	100	9.7	55.7
Verb+Prep	95.2	50.0	65.6	95.2	25.3	40.0
Infinitive	95.7	46.8	62.9	100	25.5	40.6
Other	0	0	0	0	0	0
All	**88.3**	**45.2**	**59.8**	86.6	23.2	36.6

Table 2: Open Extraction by Relation Category. O-CRF outperforms O-NB, obtaining nearly double its recall and increased precision. O-CRF's gains are partly due to its lower false positive rate for relationships categorized as "Other."

these numeric features, H-CRF uses a subset of the base features used by O-CRF and R1-CRF. At each given position i between e_1 and e_2, the presence of the word observed at i as a feature, as well as the presence of the part-of-speech-tag at i.

5 Experimental Results

The following experiments demonstrate the benefits of Open IE for two tasks: open extraction and targeted extraction.

Section 5.1, assesses the ability of O-CRF to locate instances of relationships when the number of relationships is large and their identity is unknown. We show that without any relation-specific input, O-CRF extracts binary relationships with high precision and a recall that nearly doubles that of O-NB.

Sections 5.2 and 5.3 compare O-CRF to traditional and hybrid RE when the goal is to locate instances of a small set of known target relations. We find that while single-relation extraction, as embodied by R1-CRF, achieves comparatively higher levels of recall, it takes hundreds, and sometimes thousands, of labeled examples *per relation*, for R1-CRF to approach the precision obtained by O-CRF, which is self-trained without any relation-specific input. We also show that the combination of unlexicalized, open extraction in O-CRF and lexicalized, supervised extraction in R1-CRF improves precision and F-measure compared to a standalone RE system.

5.1 Open Extraction

This section contrasts the performance of O-CRF with that of O-NB on an Open IE task, and shows that O-CRF achieves both double the recall and increased precision relative to O-NB. For this experiment, we used the set of 500 sentences[3] described in Section 2. Both IE systems were designed and trained prior to the examination of the sample sentences; thus the results on this sentence sample provide a fair measurement of their performance.

While the TEXTRUNNER system was previously found to extract over 7.5 million tuples from a corpus of 9 million Web pages, these experiments are the first to assess its true recall over a known set of relational tuples. As reported in Table 2, O-CRF extracts relational tuples with a precision of 88.3% and a recall of 45.2%. O-CRF achieves a relative gain in F1 of 63.4% over the O-NB model employed by TEXTRUNNER, which obtains a precision of 86.6% and a recall of 23.2%. The recall of O-CRF nearly doubles that of O-NB.

O-CRF is able to extract instances of the four most frequently observed relation types – Verb, Noun+Prep, Verb+Prep and Infinitive. Three of the four remaining types – Modifier, Coordinate$_n$ and Coordinate$_v$ – which comprise only 8% of the sample, are not handled due to simplifying assumptions made by both O-CRF and O-NB that tokens indicating a relation occur between entity mentions in the sentence.

5.2 O-CRF vs. R1-CRF Extraction

To compare performance of the extractors when a small set of target relationships is known in advance, we used labeled data for four different relations – corporate acquisitions, birthplaces, inventors of products and award winners. The first two datasets were collected from the Web, and made available by Bunescu and Mooney (2007). To augment the size of our corpus, we used the same technique to collect data for two additional relations, and manually labelled positive and negative instances by hand over all collections. For each of the four relations in our collection, we trained R1-CRF from labeled training data, and ran each of R1-CRF and O-CRF over the respective test sets, and compared the precision and recall of all tuples output by each system.

[3] Available at http://www.cs.washington.edu/research/knowitall/hlt-naacl08-data.txt

	O-CRF		R1-CRF		
Relation	**P**	**R**	**P**	**R**	**Train Ex**
Acquisition	75.6	19.5	67.6	69.2	3042
Birthplace	90.6	31.1	92.3	64.4	1853
InventorOf	88.0	17.5	81.3	50.8	682
WonAward	62.5	15.3	73.6	52.8	354
All	**75.0**	18.4	73.9	**58.4**	5930

Table 3: Precision (P) and Recall (R) of O-CRF and R1-CRF.

	O-CRF		R1-CRF		
Relation	**P**	**R**	**P**	**R**	**Train Ex**
Acquisition	75.6	19.5	67.6	69.2	3042*
Birthplace	90.6	31.1	92.3	53.3	600
InventorOf	88.0	17.5	81.3	50.8	682*
WonAward	62.5	15.3	65.4	61.1	50
All	**75.0**	18.4	70.17	**60.7**	>4374

Table 4: For 4 relations, a minimum of 4374 hand-tagged examples is needed for R1-CRF to approximately match the precision of O-CRF for each relation. A "*" indicates the use of all available training data; in these cases, R1-CRF was unable to match the precision of O-CRF.

Table 3 shows that from the start, O-CRF achieves a high level of precision – 75.0% – without any relation-specific data. Using labeled training data, the R1-CRF system achieves a slightly lower precision of 73.9%.

Exactly how many training examples per relation does it take R1-CRF to achieve a comparable level of precision? We varied the number of training examples given to R1-CRF, and found that in 3 out of 4 cases it takes hundreds, if not thousands of labeled examples for R1-CRF to achieve acceptable levels of precision. In two cases – acquisitions and inventions – R1-CRF is unable to match the precision of O-CRF, even with many labeled examples. Table 4 summarizes these findings.

Using labeled data, R1-CRF obtains a recall of 58.4%, compared to O-CRF, whose recall is 18.4%. A large number of false negatives on the part of O-CRF can be attributed to its lack of lexical features, which are often crucial when part-of-speech tagging errors are present. For instance, in the sentence, "Yahoo To Acquire Inktomi", "Acquire" is mistaken for a proper noun, and sufficient evidence of the existence of a relationship is absent. The lexicalized R1-CRF extractor is able to recover from this error; the presence of the word "Acquire" is enough to recognize the positive instance, despite the incorrect part-of-speech tag.

	R1-CRF			Hybrid		
Relation	**P**	**R**	**F1**	**P**	**R**	**F1**
Acquisition	67.6	69.2	68.4	76.0	67.5	71.5
Birthplace	93.6	64.4	76.3	96.5	62.2	75.6
InventorOf	81.3	50.8	62.5	87.5	52.5	65.6
WonAward	73.6	52.8	61.5	75.0	50.0	60.0
All	73.9	**58.4**	65.2	**79.2**	56.9	**66.2**

Table 5: A hybrid extractor that uses O-CRF improves precision for all relations, at a small cost to recall.

Another source of recall issues facing O-CRF is its ability to discover synonyms for a given relation. We found that while RESOLVER improves the relative recall of O-CRF by nearly 50%, O-CRF locates fewer synonyms per relation compared to its lexicalized counterpart. With RESOLVER, O-CRF finds an average of 6.5 synonyms per relation compared to R1-CRF's 16.25.

In light of our findings, the relative tradeoffs of open versus traditional RE are as follows. Open IE automatically offers a high level of precision without requiring manual labor per relation, at the expense of recall. When relationships in a corpus are not known, or their number is massive, Open IE is essential for RE. When higher levels of recall are desirable for a small set of target relations, traditional RE is more appropriate. However, in this case, one must be willing to undertake the cost of acquiring labeled training data for each relation, either via a computational procedure such as bootstrapped learning or by the use of human annotators.

5.3 Hybrid Extraction

In this section, we explore the performance of H-CRF, an ensemble-based extractor that learns to perform RE for a set of known relations based on the individual behaviors of O-CRF and R1-CRF.

As shown in Table 5, the use of O-CRF as part of H-CRF, improves precision from 73.9% to 79.2% with only a slight decrease in recall. Overall, F1 improved from 65.2% to 66.2%.

One disadvantage of a stacking-based hybrid system is that labeled training data is still required. In the future, we would like to explore the development of hybrid systems that leverage Open IE methods, like O-CRF, to reduce the number of training examples required per relation.

6 Related Work

TEXTRUNNER, the first Open IE system, is part of a body of work that reflects a growing interest in avoiding relation-specificity during extraction. Sekine (2006) developed a paradigm for "on-demand information extraction" in order to reduce the amount of effort involved when porting IE systems to new domains. Shinyama and Sekine's "preemptive" IE system (2006) discovers relationships from sets of related news articles.

Until recently, most work in RE has been carried out on a per-relation basis. Typically, RE is framed as a binary classification problem: Given a sentence S and a relation R, does S assert R between two entities in S? Representative approaches include (Zelenko et al., 2003) and (Bunescu and Mooney, 2005), which use support-vector machines fitted with language-oriented kernels to classify pairs of entities. Roth and Yih (2004) also described a classification-based framework in which they jointly learn to identify named entities and relations.

Culotta *et al.* (2006) used a CRF for RE, yet their task differs greatly from open extraction. RE was performed from biographical text in which the topic of each document was known. For every entity found in the document, their goal was to predict what relation, if any, it had relative to the page topic, from a set of given relations. Under these restrictions, RE became an instance of entity labeling, where the label assigned to an entity (*e.g. Father*) is its relation to the topic of the article.

Others have also found the stacking framework to yield benefits in the context of IE. Freitag (2000) used linear regression to model the relationship between the confidence of several inductive learning algorithms – rote learning, Naive Bayes, grammatical inference and a relational rule learning – and the probability that a prediction is correct. Over three different document collections, the combined method yielded improvements over the best individual learner for all but one relation. The efficacy of ensemble-based methods for extraction was further investigated by (Sigletos et al., 2005), who experimented with combining the outputs of a rule-based learner, a Hidden Markov Model and a wrapper-induction algorithm in five different domains. Of a variety ensemble-based methods, stacking proved to consistently outperform the best base-level system, obtaining more precise results at the cost of somewhat lower recall. (Feldman et al., 2005) demonstrated that a hybrid extractor composed of a statistical and knowledge-based models outperform either in isolation.

7 Conclusions and Future Work

Our experiments have demonstrated the promise of relation-independent extraction using the Open IE paradigm. We have shown that binary relationships can be categorized using a compact set of lexico-syntactic patterns, and presented O-CRF, a CRF-based Open IE system that can extract different relationships with a precision of 88.3% and a recall of 45.2%[4]. Open IE is essential when the number of relationships of interest is massive or unknown.

Traditional IE is more appropriate for "targeted extraction" when the number of relations of interest is small and one is willing to incur the cost of acquiring labeled training data. Compared to traditional IE, the recall of our Open IE system is admittedly lower. However, even in a targeted extraction scenario, Open IE can still be used to reduce the number of hand-labeled examples necessary. As Table 4 shows, numerous hand-labeled examples (numbers ranging from 50 for one relation to over 3,000 for anther) are necessary to match the precision of O-CRF.

In the future, O-CRF's recall may be improved by enhancements to its ability to locate the various ways in which a given relation is expressed. We also plan to explore the capacity of Open IE to automatically provide labeled training data, when traditional relation extraction is a more appropriate choice.

[4] The TEXTRUNNER Open IE system now indexes extractions found by O-CRF from millions of Web pages, and is located at http://www.cs.washington.edu/research/textrunner

Acknowledgments

This research was supported in part by NSF grants IIS-0535284 and IIS-0312988, ONR grant N00014-08-1-0431 as well as gifts from Google, and carried out at the University of Washington's Turing Center. Doug Downey, Stephen Soderland and Dan Weld provided helpful comments on previous drafts.

References

E. Agichtein and L. Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Procs. of the Fifth ACM International Conference on Digital Libraries*.

M. Banko, M. Cararella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the web. In *Procs. of IJCAI*.

S. Brin. 1998. Extracting Patterns and Relations from the World Wide Web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, pages 172–183, Valencia, Spain.

R. Bunescu and R. Mooney. 2005. Subsequence kernels for relation extraction. In *In Procs. of Neural Information Processing Systems*.

R. Bunescu and R. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proc. of ACL*.

A. Culotta and A. McCallum. 2004. Confidence estimation for information extraction. In *Procs of HLT/NAACL*.

A. Culotta, A. McCallum, and J. Betz. 2006. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Procs of HLT/NAACL*, pages 296–303.

P. Domingos. 1996. Unifying instance-based and rule-based induction. *Machine Learning*, 24(2):141–168.

O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

R. Feldman, B. Rosenfeld, and M. Fresko. 2005. Teg - a hybrid approach to information extraction. *Knowledge and Information Systems*, 9(1):1–18.

D. Freitag. 2000. Machine learning for information extraction in informal domains. *Machine Learning*, 39(2-3):169–202.

R. Girju, A. Badulescu, and D. Moldovan. 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1).

M. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Procs. of the 14th International Conference on Computational Linguistics*, pages 539–545.

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *ACL*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Procs. of ICML*.

A. McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

A. E. Monge and C. P. Elkan. 1996. The field matching problem: Algorithms and applications. In *Procs. of KDD*.

E. Riloff and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-level Boot-strapping. In *Procs. of AAAI-99*, pages 1044–1049.

D. Roth and W. Yih. 2004. A linear progamming formulation for global inference in natural language tasks. In *Procs. of CoNLL*.

S. Sekine. 2006. On-demand information extraction. In *Proc. of COLING*.

Y. Shinyama and S. Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proc. of the HLT-NAACL*.

G. Sigletos, G. Paliouras, C. D. Spyropoulos, and M. Hatzopoulos. 2005. Combining infomation extraction systems using voting and stacked generalization. *Journal of Machine Learning Research*, 6:1751,1782.

R. Snow, D. Jurafsky, and A. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17*. MIT Press.

K.M. Ting and I. H. Witten. 1999. Issues in stacked generalization. *Artificial Intelligence Research*, 10:271–289.

D. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–260.

A. Yates and O. Etzioni. 2007. Unsupervised resolution of objects and relations on the web. In *Procs of NAACL/HLT*.

D. Zelenko, C. Aone, and A. Richardella. 2003. Kernel methods for relation extraction. *JMLR*, 3:1083–1106.

B. Zenko and S. Dzeroski. 2002. Stacking with an extended set of meta-level attributes and mlr. In *Proc. of ECML*.